In the Claims:

1-9.   (Canceled)

---

10.   (Currently Amended) A device for enabling an executing entity of a host system to execute code, comprising:

(i)   a non-executable memory component, for storing the code;

(ii)   at least one executable memory component, each said executable memory component for presenting at least a portion of said stored code to the executing entity in a manner that enables the executing entity to execute said portion of said stored code directly from said each executable memory <u>via a standard memory interface</u>, each said executable memory component receiving said stored code ~~directly~~ from said non-executable memory component <u>via an interface separate from said standard memory interface</u>; and

(iii)   a mechanism for ~~guaranteeing~~ <u>indicating</u> availability, in one of said at least one executable memory component, of code requested by the executing entity.

11.   (Original) The device of claim 10, wherein said non-executable memory component and said at least one executable memory component are separate from the host system.

12.   (Canceled)

13.   (Original) The device of claim 10, comprising a plurality of said executable memory components, such that while one said executable memory component is presenting a first said at least portion of said stored code to the executing entity, a second said at least portion of said stored code is being downloaded to another said executable memory component.

14.   (Original) The device of claim 10, wherein each said at least one executable memory component is too small to accommodate all of the code at once.

15.   (Currently Amended) A method of executing code, comprising the steps of:

(a)   storing the code in a non-executable memory component;

(b)   downloading only a first portion of the code from said non-executable memory component to a first executable memory component;

(c)   executing said downloaded code, by an executing entity of a host system, directly from said first executable memory component <u>via a standard memory interface</u>, said first executable memory component being separate from said host system;

(d)   subsequent to said downloading, requesting code to be executed, by said executing entity; and

(e)   if said requested code is outside of said downloaded first portion of the code:

(i)   downloading a second portion of the code, including said requested code, from said non-executable memory component to said first executable memory component; and

3

(ii)   during said downloading of said second portion of the code, suspending activity of said executing entity;

wherein all said downloading of said portions of said code from said non-executable memory component to said first executable memory component is via an interface separate from said standard memory interface.

16-18.   (Canceled)

19.   (Original) The method of claim 15, wherein said suspending includes supplying a busy signal to said executing entity.

20.   (Original) The method of claim 15, further comprising the steps of:

(f)   downloading a third portion of the code to a second executable memory component; and

(g)   executing said downloaded third portion of the code, by said executing entity.

21.   (Original) The method of claim 20, wherein said second executable memory component is separate from said host system.

22.   (Original) The method of claim 21, wherein said executing entity executes said downloaded third portion of the code directly from said second executable memory component.

23.   (New) The device of claim 13, wherein said executable memory components are physically separate executable memory components.

24.   (New) The method of claim 20, wherein said executable memory components are physically separate executable memory components.

25.   (New) A device for enabling an executing entity of a host system to execute code, comprising:

(i)   a non-executable memory component, for storing the code;

(ii)   at least one executable memory component, each said executable memory component for presenting at least a portion of said stored code to the executing entity in a manner that enables the executing entity to execute said portion of said stored code directly from said each executable memory, each said executable memory component receiving said stored code ~~directly~~ from said non-executable memory component <u>independently of said executing entity</u>; and

(iii) a mechanism for ~~guaranteeing~~ indicating availability, in one of said at least one executable memory component, of code requested by the executing entity.